Федеральное государственное бюджетное образовательное учреждение

высшего образования

Российский экономический университет имени Г.В. Плеханова

Московский приборостроительный техникум

Специальность: 09.02.03 Программирование в компьютерных системах

ПМ 02 Разработка и администрирование баз данных

МДК 02.02 Технология разработки и защиты баз данных

Отчёт

МПТ.МДК.02.02.ТРиЗБД.П-2-17

|  |  |
| --- | --- |
| Проверил | Выполнил |
| Щаников И.М. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Студент группы |
| «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г. | П-2-17 |
| Оценка: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Алпатов Н.С. |
| (\_\_\_\_\_\_\_\_\_)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г. |

2020 г

СОДЕРЖАНИЕ

[1. ОПИСАНИЕ ПРОБЛЕМНОЙ ЗОНЫ 3](#_Toc56010134)

[1.1. Описание предметной области 3](#_Toc56010135)

[1.2. Описание подсистем 3](#_Toc56010136)

[1.3. Описание сущностей 5](#_Toc56010137)

[2. ПРОЕКТИРОВАНИЕ 6](#_Toc56010138)

[2.1. Модель данных 6](#_Toc56010139)

[2.1.1. Инфологическая модель данных 6](#_Toc56010140)

[2.1.2. Даталогическая модель данных 7](#_Toc56010141)

[2.2. Описание объектов 8](#_Toc56010142)

[2.2.1. Ограничения 8](#_Toc56010143)

[2.2.2. Хранимые процедуры 10](#_Toc56010144)

[2.2.3. Функции 12](#_Toc56010145)

[2.3.4. Представления 12](#_Toc56010146)

[2.3. Описание словаря данных 15](#_Toc56010147)

[3. РЕАЛИЗАЦИЯ 18](#_Toc56010148)

[3.1. Таблицы базы данных 18](#_Toc56010149)

[3.2. Хранимые процедуры базы данных 30](#_Toc56010150)

[3.3. Функции базы данных 41](#_Toc56010151)

[3.4. Представления базы данных 43](#_Toc56010152)

[4. ТЕСТИРОВАНИЕ 46](#_Toc56010153)

[4.1. Сценарии 46](#_Toc56010154)

[4.1.1. Таблицы 46](#_Toc56010155)

[4.1.2. Хранимые процедуры 52](#_Toc56010156)

[4.1.3. Функции 54](#_Toc56010157)

[4.1.4. Представления 55](#_Toc56010158)

[4.2. Результаты тестирования 58](#_Toc56010159)

[4.2.1. Таблицы 58](#_Toc56010160)

[4.2.2. Хранимые процедуры 61](#_Toc56010161)

[4.2.3. Функции 65](#_Toc56010162)

[4.2.4. Представления 68](#_Toc56010163)

[ЗАКЛЮЧЕНИЕ 71](#_Toc56010164)

[СПИСОК ИСПОЛЬЗУЕМЫХ МАТЕРИАЛОВ 72](#_Toc56010165)

# ОПИСАНИЕ ПРОБЛЕМНОЙ ЗОНЫ

## Описание предметной области

Предметной областью базы данных является продажа автомобилей в автосалоне. Автосалон − это некий аналог магазина, в котором представлены существующие образцы новых автомобилей, а также имеются комплектующие для автомобилей. Данная база предназначена для автосалонов. Потребителями будут сотрудники автосалона, они будут вести продажу на основе данной базы.

Имеются две подсистемы: учет автомобилей и учет комплектующих. Эти подсистемы связаны. Клиент может заказать отдельные комплектующие для автомобиля, которые он выберет для покупки.

## Описание подсистем

Порядок выполнения заказа на автомобиль и на комплектующие аналогичный друг другу.

Первым делом на вход поступает заказ клиента, на основе законодательства РФ, лицензии на продажу, каталога автомобилей.

Порядок выполнения заказа автомобиля:

План закупок автомобилей выполняют отдел бухгалтерии, отдел по закупке автомобилей и сам автосалон. На вход поступает заказ клиента и текущий поставщик автомобилей. Формируется план, на основе которого оформляется договор с поставщиками. Это выполняет отдел по закупке автомобилей, юридический отдел и сам автосалон. Далее на основе договора формируется каталог автомобилей. Это действие выполняют автосалон и отдел рекламы. Следующий этап − сбыт автомобилей. Выполняется на основе сформированного каталога, лицензии на продажу, законодательства РФ. Выполняет это отдел бухгалтерии и отдела сбыта автомобилей. На выходе − выполненный заказ на автомобиль.

Порядок выполнения заказа комплектующих:

План закупок комплектующих выполняют отдел бухгалтерии, отдел по закупке комплектующих и сам автосалон. На вход поступает заказ клиента и текущий поставщик комплектующих. Формируется план, на основе которого оформляется договор с поставщиками. Это выполняет отдел по закупке комплектующих, юридический отдел и сам автосалон. Далее на основе договора формируется каталог комплектующих. Это действие выполняют автосалон и отдел рекламы. Следующий этап − сбыт комплектующих. Выполняется на основе сформированного каталога, лицензии на продажу, законодательства РФ. Выполняет это отдел бухгалтерии и отдела сбыта комплектующих. На выходе − выполненный заказ на комплектующие.

## Описание сущностей

В подсистему учет автомобилей будут входить следующие сущности:

− Автомобиль (код автомобиля, код поставщика FK, название автомобиля, VIN, код типа автомобиля FK, код характеристики FK).

− Тип автомобиля (код типа автомобиля, название типа автомобиля).

− Характеристики типов автомобилей (код характеристики типа автомобилей, код характеристики FK, код типа автомобиля FK).

− ЗаказАвтомобиль (код ЗаказаАвтомобиля, код заказа FK, код автомобиля FK).

В подсистему учет комплектующих будут входить следующие сущности:

− Товар (код товара, код поставщика FK, код типа FK, название товара, артикул, код характеристики FK).

− Тип товара (код типа, название типа товара).

− Характеристики типов товара (код характеристики типа, код типа FK, код характеристики FK).

Связующими сущностями между подсистемами будут являться:

− Характеристики (код характеристики, название характеристики, цена).

− Поставщик (код поставщика, название).

Дополнительные сущности:

− Сотрудник (код сотрудника, фамилия, имя, отчество, код отдела FK).

− Отдел (код отдела, наименование отдела).

− Клиент (код клиента, фамилия, имя, отчество).

− Заказ (код заказа, код сотрудника FK, код клиента FK, дата заказа).

− ЗаказТовар (код ЗаказаТовара, код заказа FK, код товара FK).

# ПРОЕКТИРОВАНИЕ

## Модель данных

### Инфологическая модель данных

На Рисунке 1 изображена инфологическая модель базы данных.

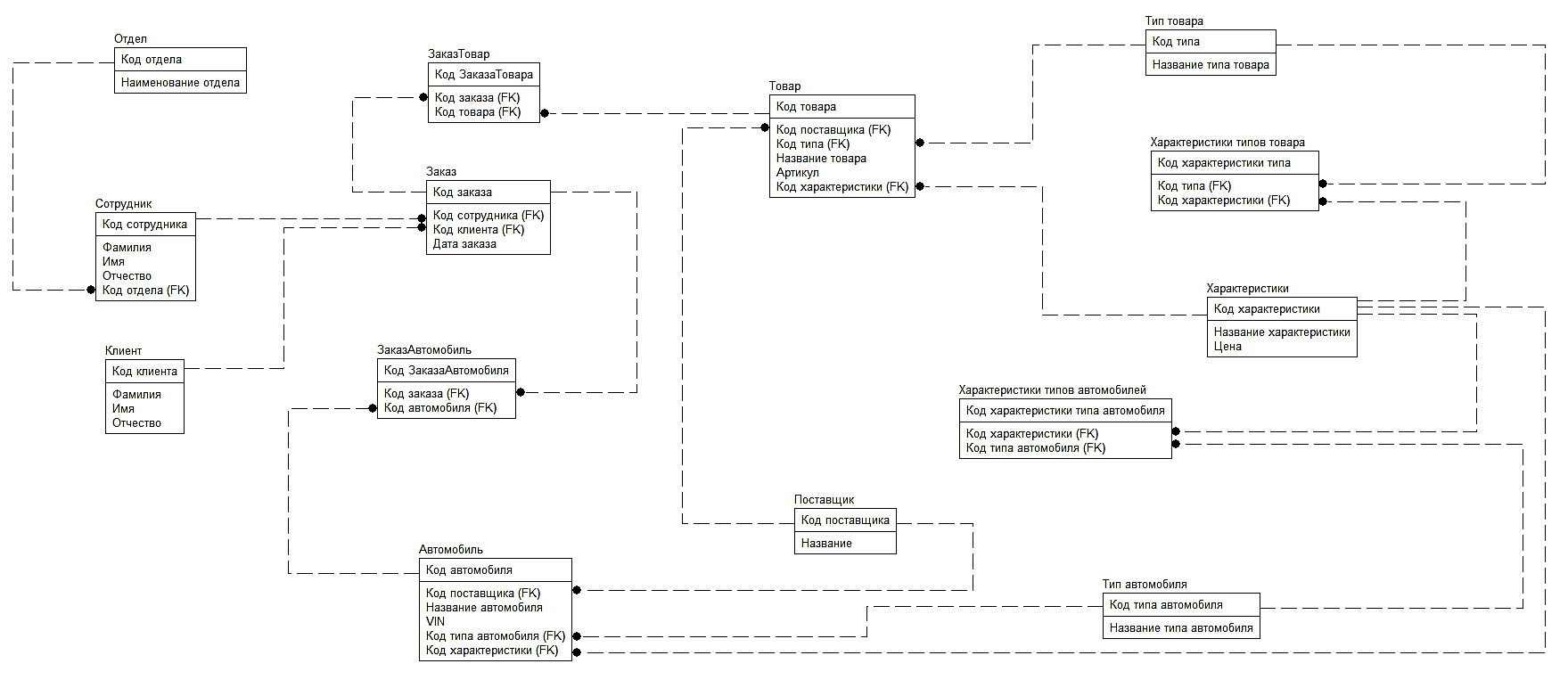


Рисунок 1 − Инфологическая модель базы данных.

### Даталогическая модель данных

На Рисунке 1 изображена инфологическая модель базы данных.

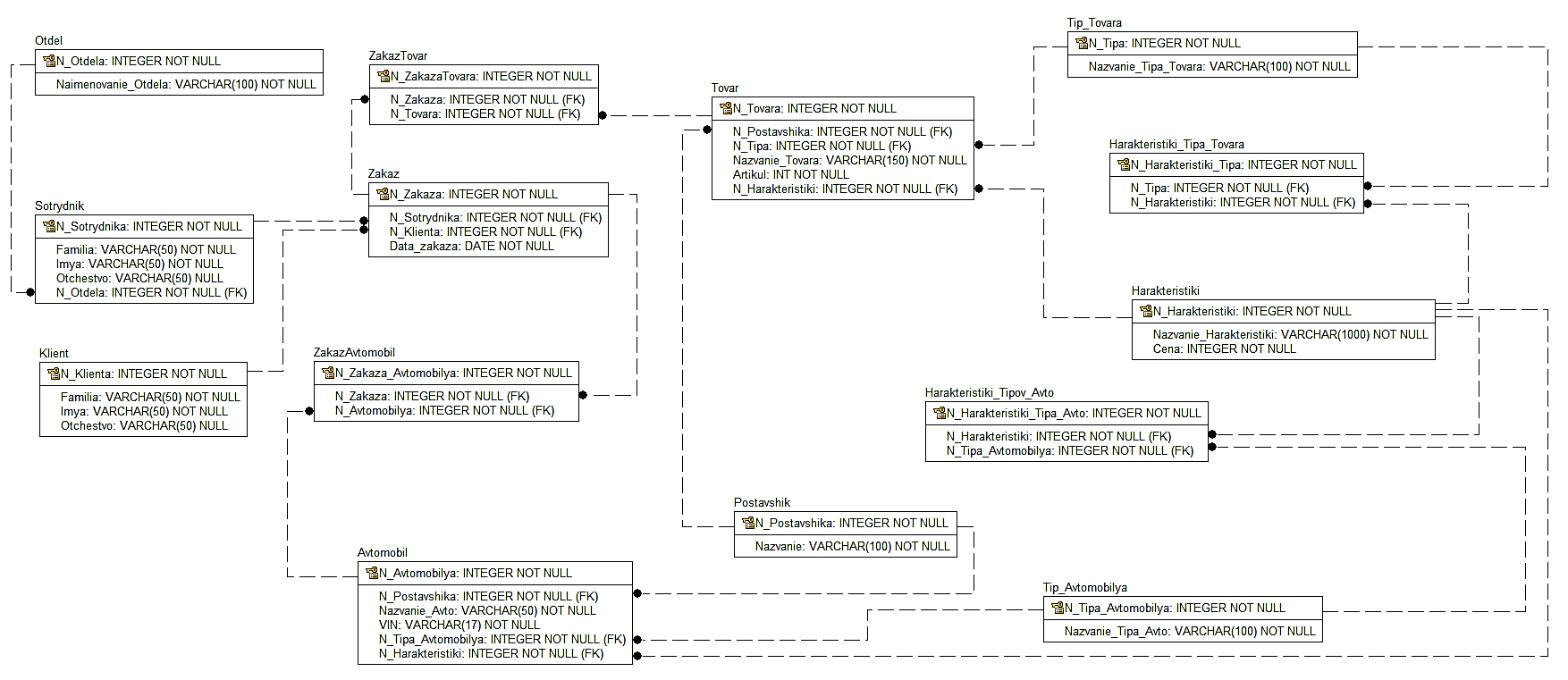


Рисунок 2 − Даталогическая модель базы данных.

## Описание объектов

### Ограничения

В Таблице 1 представлены ограничения для атрибутов сущностей.

Таблица 1 – Ограничения.

| Таблица | Поле | Ограничения | Тип |
| --- | --- | --- | --- |
| Clients | id | [0-9], not null | INTEGER |
| Surname | [А-Я, а-я, A-Z, a-z], {1, 255}, not null | VARCHAR |
| Name | [А-Я, а-я, A-Z, a-z], {1, 255}, not null | VARCHAR |
| Patronomyc | [А-Я, а-я, A-Z, a-z], {1, 255}, not null | VARCHAR |
| The\_passport\_series | [0-9], not null | INTEGER |
| Passport\_number | [0-9], not null | INTEGER |
| Telephone | [А-Я, а-я, A-Z, a-z], {11}, not null | VARCHAR |
| Date\_of\_birthday | [01.01.0001 - 31.12.9999], [0-9], not null | DATE |
| Orders | id | [0-9], not null | INTEGER |
| Discription | [А-Я, а-я, A-Z, a-z], {1, 255}, not null | VARCHAR |
| Spare\_parts\_for\_repairs | [А-Я, а-я, A-Z, a-z], {1, 255}, not null | VARCHAR |
| Car\_number | [А-Я, а-я, A-Z, a-z], {1, 255}, not null | VARCHAR |
| Car\_model | [А-Я, а-я, A-Z, a-z], {1, 255}, not null | INTEGER |
| Finnaly\_cost | [0-9], not null | INTEGER |
| Service\_start\_date | [01.01.0001 - 31.12.9999], [0-9], not null | DATE |
| id\_Worker | [0-9], not null | INTEGER |
| id\_Ongoing\_operation | [0-9], not null | INTEGER |
| id\_Owner | [0-9], not null | INTEGER |
| Workers | id | [0-9], not null | INTEGER |
| Surname | [А-Я, а-я, A-Z, a-z], {1, 255}, not null | VARCHAR |
| Name | [А-Я, а-я, A-Z, a-z], {1, 255}, not null | VARCHAR |
| Patronomyc | [[А-Я, а-я, A-Z, a-z], {1, 255}, not null | VARCHAR |
| The\_passport\_series | [0-9], not null | INTEGER |
| Passport\_number | [0-9], not null | INTEGER |
| Residence | [А-Я, а-я, A-Z, a-z], {1, 255}, not null | VARCHAR |
| Telephone | [А-Я, а-я, A-Z, a-z], {11}, not null | VARCHAR |
| Id\_Positions | [0-9], not null | INTEGER |
| Positions | id | [0-9], not null | INTEGER |
| Salary | [0-9], not null | INTEGER |
| Name | [А-Я, а-я, A-Z, a-z], {1, 255}, not null | VARCHAR |
| Duties | [А-Я, а-я, A-Z, a-z], {1, 255}, not null | VARCHAR |
| Description | [А-Я, а-я, A-Z, a-z], {1, 255}, not null | VARCHAR |
| Products\_in\_stock | Id | [0-9], not null | INTEGER |
| Name | [А-Я, а-я, A-Z, a-z], {1, 255}, not null | VARCHAR |
| Manufacturer | [А-Я, а-я, A-Z, a-z], {1, 255}, not null | VARCHAR |
| Description | [А-Я, а-я, A-Z, a-z], {1, 255}, not null | VARCHAR |
| Purchase\_value | [0-9], not null | INTEGER |
| Cost\_for\_sale | [0-9], not null | INTEGER |
| Count | [0-9], not null | INTEGER |
| Do\_i\_need\_to\_re\_oreder | [А-Я, а-я, A-Z, a-z], {1, 255}, not null | VARCHAR |
| Price\_list | id | [0-9], not null | INTEGER |
| Name\_of\_the\_operation | [А-Я, а-я, A-Z, a-z], {1, 255}, not null | VARCHAR |
| Price | [0-9], not null | INTEGER |
| Id\_Product\_in\_stock | [0-9], not null | INTEGER |

### Хранимые процедуры

* CLIENT\_INSERT − Добавление клиента
* CLIENT \_UPDATE − Обновление клиента
* CLIENT \_DELETE − Удаление клиента
* ORDERS\_INSERT − Добавление заказа
* ORDERS \_UPDATE − Обновление заказа
* ORDERS\_DELETE − Удаление заказа
* WORKERS\_INSERT − Добавление сотрудника
* WORKERS\_UPDATE − Обновление сотрудника
* WORKERS\_DELETE − Удаление сотрудника
* POSITIONS\_INSERT − Добавление должности
* POSITIONS\_UPDATE − Обновление должности
* POSITIONS \_DELETE − Удаление должности
* PRICE\_LIST\_INSERT − Добавление услуги
* PRICE\_LIST\_UPDATE − Обновление услуги
* PRICE\_LIST\_DELETE − Удаление услуги
* PRODUCTS\_IN\_STOCK\_INSERT − Добавление товара на складе
* PRODUCTS\_IN\_STOCK\_UPDATE − Обновление товара на складе
* PRODUCTS\_IN\_STOCK \_DELETE − Удаление товара на складе

### Функции

* COUNT\_OF\_WORKERS − Подсчет количества сотрудников
* COUNT\_OF\_CLIENTS − Подсчет количества клиентов
* COUNT\_OF\_ ORDERS− Подсчет количества заказов

### 2.3.4. Представления

* O\_W\_C\_P − Вывод полной информации о заказе и сотруднике который над ним работает.

Таблицы, участвующие в представлении и их поля.

1. ORDERS(DISCRIPTION , SPARE\_PARTS\_FOR\_REPAIRS , CAR\_NUMBER, CAR\_MODEL, FINNALY\_COST , SERVICE\_START\_DATE)
2. WORKERS(NAME , TELEPHONE)
3. CLIENTS(NAME)
4. POSITIONS(NAME\_OF\_THE\_OPERATION, PRICE)

По итогу получается таблица со следующими полями: (DISCRIPTION, SPARE\_PARTS\_FOR\_REPAIRS, CAR\_NUMBER, CAR\_MODEL, FINNALY\_COST, SERVICE\_START\_DATE, NAME, TELEPHONE, NAME, NAME\_OF\_THE\_OPERATION, PRICE)

* PRICE\_LIST\_PRODUCTS\_IN\_STOCK – Вывод информации о прайс листе и товарах использующихся при определенной услуге.

Таблицы, участвующие в представлении и их поля.

1. PRICE\_LIST(NAME\_OF\_THE\_OPERATION, PRICE)
2. PRODUCTS\_IN\_STOCK(NAME,DESCRIPTION,MANUFACTURER)

По итогу получается таблица со следующими полями: (NAME\_OF\_THE\_OPERATION, PRICE, NAME, DESCRIPTION, MANUFACTURER)

* WORKERS\_POSITIONS– Вывод информации о работниках и занимаемыми ими должностях.

Таблицы, участвующие в представлении и их поля.

1. POSITIONS(NAME , SALARY, DUTIES, DESCRIPTION)
2. WORKERS(NAME, RESIDENCE, TELEPHONE)

По итогу получается таблица со следующими полями: (NAME , SALARY, DUTIES, DESCRIPTION, NAME, RESIDENCE, TELEPHONE)

## Описание словаря данных

В данном разделе в Таблице 2 представлен словарь данных.

Таблица 2 – Словарь данных

|  |  |  |  |
| --- | --- | --- | --- |
| Таблица | | Логическое название | Физическое название |
| Клиенты | Clients |
| Логическое название | Физическое название | Тип данных | Ограничения |
| Номер клиента (PK) | id | integer | not null |
| Фамилия | Surname | varchar | {1, 255}, not null |
| Имя | Name | varchar | {1, 255}, not null |
| Отчество | Patronomyc | varchar | {1, 255}, not null |
| Серия паспорта | The\_passport\_series | integer | not null |
| Номер паспорта | Passport\_number | integer | not null |
| Телефон | Telephone | varchar | {11}, not null |
| Дата рождения | Date\_of\_birthday | date | not null |
| Таблица | | Логическое название | Физическое название |
| Заказы | Orders |
| Логическое название | Физическое название | Тип данных | Ограничения |
| Номер заказа (PK) | id | integer | not null |
| Описание | Discription | varchar | {1, 255}, not null |
| Запчасти для ремонта | Spare\_parts\_for\_repairs | varchar | {1, 255}, not null |
| Номер машины | Car\_number | varchar | {1, 255}, not null |
| Модель машины | Car\_model | varchar | {1, 255}, not null |
| Финальная стоимость | Finnaly\_cost | integer | not null |
| Дата начала ремонта | Service\_start\_date | date | not null |
| Номер сотрудника | id\_Worker | integer | not null |
| Номер услуги | id\_Ongoing\_operation | integer | not null |
| Номер владельца | id\_Owner | integer | not null |
| Таблица | | Логическое название | Физическое название |
| Работник | Workers |
| Логическое название | Физическое название | Тип данных | Ограничения |
| Номер работника (PK) | id | integer | not null |
| Фамилия | Surname | varchar | {1, 255}, not null |
| Имя | Name | varchar | {1, 255}, not null |
| Отчество | Patronomyc | varchar | {1, 255}, not null |
| Серия паспорта | The\_passport\_series | integer | not null |
| Номер паспорта | Passport\_number | integer | not null |
| Отчество | Residence | varchar | {1, 255}, not null |
| Телефон | Telephone | varchar | {11}, not null |
| Номер должности | Id\_Positions | integer | Not null |
| Таблица | | Логическое название | Физическое название |
| Должности | Positions |
| Логическое название | Физическое название | Тип данных | Ограничения |
| Номер должности (PK) | id | integer | not null |
| Зарплата | Salary | integer | not null |
| Название должности | Name | varchar | not null |
| Обязанности | Duties | varchar | {10}, not null |
| Описание должности | Description | varchar | {1, 255}, not null |
| Таблица | | Логическое название | Физическое название |
| Товары на складе | Products\_in\_stock |
| Логическое название | Физическое название | Тип данных | Ограничения |
| Номер детали (PK) | id | integer | not null |
| Название детали | Name | varchar | {1, 255}, not null |
| Производитель | Manufacturer | varchar | {1, 255}, not null |
| Описание | Description | varchar | {1, 255}, not null |
| Цена закупки | Purchase\_value | integer | not null |
| Цена продажи | Cost\_for\_sale | integer | not null |
| Количество | Count | integer | not null |
| Нужно ли дозаказать | Do\_i\_need\_to\_re\_oreder | varchar | {1, 255}, not null |
| Таблица | | Логическое название | Физическое название |
| Прайс лист | Price\_list |
| Логическое название | Физическое название | Тип данных | Ограничения |
| Номер услуги (PK) | id | integer | not null |
| Название услуги | Name\_of\_the\_operation | varchar | {1, 255}, not null |
| Цена | Price | integer | not null |
| Номер запчасти на складе | Id\_Product\_in\_stock | integer | not null |

# РЕАЛИЗАЦИЯ

## Таблицы базы данных

/\*СОЗДАНИЕ ТАБЛИЦЫ "КЛИЕНТЫ" С ПОЛЯМИ "НОМЕР КЛИЕНТА(PK), ФАМИЛИЯ, ИМЯ, ОТЧЕСТВО, СЕРИЯ ПАСПОРТА, НОМЕР ПАСПОРТА, ТЕЛЕФОН, ДАТА РОЖДЕНИЯ"\*/

CREATE TABLE CLIENTS

(

ID INT NOT NULL CONSTRAINT PK\_CLIENTS PRIMARY KEY,

SURNAME VARCHAR (255) NOT NULL,

NAME VARCHAR (255) NOT NULL,

PATRONYMIC VARCHAR (255) DEFAULT '-',

THE\_PASSPORT\_SERIES INTEGER NOT NULL,

PASSPORT\_NUMBER INTEGER NOT NULL,

TELEPHONE VARCHAR (11) NOT NULL,

DATE\_OF\_BIRTHDAY DATE NOT NULL

);

/\*СОЗДАНИЕ ТАБЛИЦЫ "ТОВАРЫ НА СКЛАДЕ" И ПОЛЯМИ "НОМЕР ДЕТАЛИ (PK), НАЗВАНИЕ ДЕТАЛИ, ПРОИЗВОДИТЕЛЬ, ОПИСАНИЕ, ЦЕНА ЗАКУПКИ, ЦЕНА ПРОДАЖИ, КОЛИЧЕСТВО, НУЖНО ЛИ ДОЗАКАЗАТЬ"\*/

CREATE TABLE PRODUCTS\_IN\_STOCK

(

ID INT NOT NULL CONSTRAINT PK\_PRODUCTS\_IN\_STOCK PRIMARY KEY,

NAME VARCHAR (255) NOT NULL,

MANUFACTURER VARCHAR(255) NOT NULL,

DESCRIPTION VARCHAR (255) NOT NULL,

PURSHASE\_VALUE INTEGER NOT NULL,

COST\_FOR\_SALE INTEGER NOT NULL,

COUNT INTEGER NOT NULL,

DO\_I\_NEED\_TO\_RE\_ORDER VARCHAR (3) NOT NULL

);

/\*СОЗДАНИЕ ТАБЛИЦЫ "ДОЛЖНОСТИ" И ПОЛЯМИ "НОМЕР ДОЛЖНОСТИ (PK), ЗАРПЛАТА, НАЗВАНИЕ ДОЛЖНОСТИ, ОБЯЗАННОСТИ, ОПИСАНИЕ ДОЛЖНОСТИ"\*/

CREATE TABLE POSITIONS

(

ID INT NOT NULL CONSTRAINT PK\_POSITIONS PRIMARY KEY,

SALARY INTEGER NOT NULL,

NAME VARCHAR (255) NOT NULL,

DUTIES VARCHAR (255) NOT NULL,

DESCRIPTION VARCHAR (255) NOT NULL

);

/\*СОЗДАНИЕ ТАБЛИЦЫ "РАБОТНИКИ" И ПОЛЯМИ "НОМЕР РАБОТНИКА (PK), ФАМИЛИЯ, ИМЯ, ОТЧЕСТВО, СЕРИЯ ПАСПОРТА, НОМЕР ПАСПОРТА, ОТЧЕСТВО, ТЕЛЕФОН, НОМЕР ДОЛЖНОСТИ"\*/

CREATE TABLE WORKERS

(

ID INT NOT NULL CONSTRAINT PK\_WORKERS PRIMARY KEY,

SURNAME VARCHAR (255) NOT NULL,

NAME VARCHAR (255) NOT NULL,

PATRONYMIC VARCHAR (255) DEFAULT '-',

THE\_PASSPORT\_SERIES INTEGER NOT NULL,

PASSPORT\_NUMBER INTEGER NOT NULL,

RESIDENCE VARCHAR(255) NOT NULL,

TELEPHONE VARCHAR (11) NOT NULL,

ID\_POSITIONS INTEGER NOT NULL REFERENCES POSITIONS(ID)

);

/\*СОЗДАНИЕ ТАБЛИЦЫ "ПРАЙС ЛИСТ" И ПОЛЯМИ "НОМЕР УСЛУГИ (PK), НАЗВАНИЕ УСЛУГИ, ЦЕНА, НОМЕР ЗАПЧАСТИ НА СКЛАДЕ"\*/

CREATE TABLE PRICE\_LIST

(

ID INT NOT NULL CONSTRAINT PK\_PRICE\_LIST PRIMARY KEY,

NAME\_OF\_THE\_OPERATION VARCHAR (255) NOT NULL,

PRICE INTEGER NOT NULL,

ID\_PRODUCT\_IN\_STOCK INTEGER NOT NULL REFERENCES PRODUCTS\_IN\_STOCK(ID)

);

/\*СОЗДАНИЕ ТАБЛИЦЫ "ЗАКАЗЫ" И ПОЛЯМИ "НОМЕР ЗАКАЗА (PK), ОПИСАНИЕ, ЗАПЧАСТИ ДЛЯ РЕМОНТА, НОМЕР МАШИНЫ, МОДЕЛЬ МАШИНЫ, ФИНАЛЬНАЯ СТОИМОСТЬ, ДАТА НАЧАЛА РЕМОНТА, НОМЕР СОТРУДНИКА, НОМЕР УСЛУГИ, НОМЕР ВЛАДЕЛЬЦА"\*/

CREATE TABLE ORDERS

(

ID INT NOT NULL CONSTRAINT PK\_ORDERS PRIMARY KEY,

DISCRIPTION VARCHAR (255) NOT NULL,

SPARE\_PARTS\_FOR\_REPAIRS VARCHAR (255) NOT NULL,

CAR\_NUMBER VARCHAR (255) NOT NULL,

CAR\_MODEL VARCHAR (255) NOT NULL,

FINNALY\_COST INTEGER NOT NULL,

SERVICE\_START\_DATE DATE NOT NULL,

ID\_WORKER INTEGER NOT NULL REFERENCES WORKERS(ID),

ID\_ONGOING\_OPERATION INTEGER NOT NULL REFERENCES PRICE\_LIST(ID),

ID\_OWNER INTEGER NOT NULL REFERENCES CLIENTS(ID)

);

/\*СОЗДАНИЕ ТРИГГЕРА-АВТОИНКРЕМЕНТА ДЛЯ ТАБЛИЦЫ "КЛИЕНТЫ"\*/

CREATE SEQUENCE SQ\_CLIENTS START WITH 1;

CREATE OR REPLACE TRIGGER TG\_CLIENTS

BEFORE INSERT ON CLIENTS

FOR EACH ROW

BEGIN

SELECT SQ\_CLIENTS.NEXTVAL

INTO :NEW.ID

FROM DUAL;

END;

/\*СОЗДАНИЕ ТРИГГЕРА-АВТОИНКРЕМЕНТА ДЛЯ ТАБЛИЦЫ "ТОВАРЫ НА СКЛАДЕ"\*/

CREATE SEQUENCE SQ\_PRODUCTS\_IN\_STOCK START WITH 1;

CREATE OR REPLACE TRIGGER TG\_PRODUCTS\_IN\_STOCK

BEFORE INSERT ON PRODUCTS\_IN\_STOCK

FOR EACH ROW

BEGIN

SELECT SQ\_PRODUCTS\_IN\_STOCK.NEXTVAL

INTO :NEW.ID

FROM DUAL;

END;

/\*СОЗДАНИЕ ТРИГГЕРА-АВТОИНКРЕМЕНТА ДЛЯ ТАБЛИЦЫ "ДОЛЖНОСТИ"\*/

CREATE SEQUENCE SQ\_POSITIONS START WITH 1;

CREATE OR REPLACE TRIGGER TG\_POSITIONS

BEFORE INSERT ON POSITIONS

FOR EACH ROW

BEGIN

SELECT SQ\_POSITIONS.NEXTVAL

INTO :NEW.ID

FROM DUAL;

END;

/\*СОЗДАНИЕ ТРИГГЕРА-АВТОИНКРЕМЕНТА ДЛЯ ТАБЛИЦЫ "РАБОТНИКИ"\*/

CREATE SEQUENCE SQ\_WORKERS START WITH 1;

CREATE OR REPLACE TRIGGER TG\_WORKERS

BEFORE INSERT ON WORKERS

FOR EACH ROW

BEGIN

SELECT SQ\_WORKERS.NEXTVAL

INTO :NEW.ID

FROM DUAL;

END;

/\*СОЗДАНИЕ ТРИГГЕРА-АВТОИНКРЕМЕНТА ДЛЯ ТАБЛИЦЫ "ПРАЙС ЛИСТ"\*/

CREATE SEQUENCE SQ\_PRICE\_LIST START WITH 1;

CREATE OR REPLACE TRIGGER TG\_PRICE\_LIST

BEFORE INSERT ON PRICE\_LIST

FOR EACH ROW

BEGIN

SELECT SQ\_PRICE\_LIST.NEXTVAL

INTO :NEW.ID

FROM DUAL;

END;

/\*СОЗДАНИЕ ТРИГГЕРА-АВТОИНКРЕМЕНТА ДЛЯ ТАБЛИЦЫ "ЗАКАЗЫ"\*/

CREATE SEQUENCE SQ\_ORDERS START WITH 1;

CREATE OR REPLACE TRIGGER TG\_ORDERS

BEFORE INSERT ON ORDERS

FOR EACH ROW

BEGIN

SELECT SQ\_ORDERS.NEXTVAL

INTO :NEW.ID

FROM DUAL;

END;

## Хранимые процедуры базы данных

В данном разделе предоставлен скрипт базы данных для хранимых процедур.

/\*СОЗДАНИЕ ПРОЦЕДУРЫ ДОБАВЛЕНИЯ ДЛЯ ТАБЛИЦЫ "КЛИЕНТЫ"\*/

CREATE OR REPLACE PROCEDURE CLIENTS\_INSERT( SURNAME IN VARCHAR, NAME IN VARCHAR, PATRONYMIC IN VARCHAR, THE\_PASSPORT\_SERIES IN INT, PASSPORT\_NUMBER IN INT, TELEPHONE IN VARCHAR , DATE\_OF\_BIRTHDAY IN DATE)

IS

BEGIN

INSERT INTO CLIENTS (SURNAME, NAME, PATRONYMIC, THE\_PASSPORT\_SERIES, PASSPORT\_NUMBER, TELEPHONE, DATE\_OF\_BIRTHDAY)

VALUES (SURNAME, NAME, PATRONYMIC, THE\_PASSPORT\_SERIES, PASSPORT\_NUMBER, TELEPHONE, DATE\_OF\_BIRTHDAY);

COMMIT;

END;

/\*СОЗДАНИЕ ПРОЦЕДУРЫ УДАЛЕНИЕ ДЛЯ ТАБЛИЦЫ "КЛИЕНТЫ"\*/

CREATE OR REPLACE PROCEDURE CLIENTS\_DELETE(P\_ID IN INT)

IS

BEGIN

DELETE FROM CLIENTS

WHERE

ID = P\_ID;

COMMIT;

END;

/\*СОЗДАНИЕ ПРОЦЕДУРЫ ИЗМЕНЕНИЯ ДЛЯ ТАБЛИЦЫ "КЛИЕНТЫ"\*/

CREATE OR REPLACE PROCEDURE CLIENTS\_UPDATE(SURNAME IN VARCHAR, NAME IN VARCHAR, PATRONYMIC IN VARCHAR, THE\_PASSPORT\_SERIES IN INT, PASSPORT\_NUMBER IN INT, TELEPHONE IN VARCHAR , DATE\_OF\_BIRTHDAY IN DATE, P\_ID IN INT)

IS

BEGIN

UPDATE CLIENTS SET

SURNAME = SURNAME,

NAME = NAME,

PATRONYMIC = PATRONYMIC,

THE\_PASSPORT\_SERIES = THE\_PASSPORT\_SERIES,

PASSPORT\_NUMBER = PASSPORT\_NUMBER,

TELEPHONE = TELEPHONE,

DATE\_OF\_BIRTHDAY = DATE\_OF\_BIRTHDAY

WHERE

ID = P\_ID;

COMMIT;

END;

/\*СОЗДАНИЕ ПРОЦЕДУРЫ ДОБАВЛЕНИЕ ДЛЯ ТАБЛИЦЫ "ЗАКАЗОВ"\*/

CREATE OR REPLACE PROCEDURE ORDERS\_INSERT( DISCRIPTION IN VARCHAR, SPARE\_PARTS\_FOR\_REPAIRS IN VARCHAR, CAR\_NUMBER IN VARCHAR, CAR\_MODEL IN VARCHAR, FINNALY\_COST IN INT, SERVICE\_START\_DATE IN DATE , ID\_WORKER IN INT, ID\_ONGOING\_OPERATION IN INT, ID\_OWNER IN INT)

IS

BEGIN

INSERT INTO ORDERS (DISCRIPTION, SPARE\_PARTS\_FOR\_REPAIRS, CAR\_NUMBER, CAR\_MODEL, FINNALY\_COST, SERVICE\_START\_DATE, ID\_WORKER, ID\_ONGOING\_OPERATION, ID\_OWNER)

VALUES (DISCRIPTION, SPARE\_PARTS\_FOR\_REPAIRS, CAR\_NUMBER, CAR\_MODEL, FINNALY\_COST, SERVICE\_START\_DATE, ID\_WORKER, ID\_ONGOING\_OPERATION, ID\_OWNER);

COMMIT;

END;

/\*СОЗДАНИЕ ПРОЦЕДУРЫ УДАЛЕНИЕ ДЛЯ ТАБЛИЦЫ "ЗАКАЗОВ"\*/

CREATE OR REPLACE PROCEDURE ORDERS\_DELETE(P\_ID IN INT)

IS

BEGIN

DELETE FROM ORDERS

WHERE

ID = P\_ID;

COMMIT;

END;

/\*СОЗДАНИЕ ПРОЦЕДУРЫ ИЗМЕНЕНИЕ ДЛЯ ТАБЛИЦЫ "ЗАКАЗОВ"\*/

CREATE OR REPLACE PROCEDURE ORDERS\_UPDATE(DISCRIPTION IN VARCHAR, SPARE\_PARTS\_FOR\_REPAIRS IN VARCHAR, CAR\_NUMBER IN VARCHAR, CAR\_MODEL IN VARCHAR, FINNALY\_COST IN INT, SERVICE\_START\_DATE IN DATE , ID\_WORKER IN INT, ID\_ONGOING\_OPERATION IN INT, ID\_OWNER IN INT, P\_ID IN INT)

IS

BEGIN

UPDATE ORDERS SET

DISCRIPTION = DISCRIPTION,

SPARE\_PARTS\_FOR\_REPAIRS = SPARE\_PARTS\_FOR\_REPAIRS,

CAR\_NUMBER = CAR\_NUMBER,

CAR\_MODEL = CAR\_MODEL,

FINNALY\_COST = FINNALY\_COST,

SERVICE\_START\_DATE = SERVICE\_START\_DATE,

ID\_WORKER = ID\_WORKER,

ID\_ONGOING\_OPERATION = ID\_ONGOING\_OPERATION,

ID\_OWNER = ID\_OWNER

WHERE

ID = P\_ID;

COMMIT;

END;

/\*СОЗДАНИЕ ПРОЦЕДУРЫ ДОБАВЛЕНИЕ ДЛЯ ТАБЛИЦЫ "РАБОТНИКОВ"\*/

CREATE OR REPLACE PROCEDURE WORKERS\_INSERT( SURNAME IN VARCHAR, NAME IN VARCHAR, PATRONYMIC IN VARCHAR, PASSPORT\_NUMBER IN INT, THE\_PASSPORT\_SERIES IN INT,RESIDENCE IN VARCHAR, TELEPHONE IN VARCHAR , ID\_POSITIONS IN INT)

IS

BEGIN

INSERT INTO WORKERS (SURNAME , NAME , PATRONYMIC , PASSPORT\_NUMBER , THE\_PASSPORT\_SERIES ,RESIDENCE , TELEPHONE, ID\_POSITIONS)

VALUES (SURNAME , NAME , PATRONYMIC , PASSPORT\_NUMBER , THE\_PASSPORT\_SERIES ,RESIDENCE , TELEPHONE, ID\_POSITIONS);

COMMIT;

END;

/\*СОЗДАНИЕ ПРОЦЕДУРЫ УДАЛЕНИЕ ДЛЯ ТАБЛИЦЫ "РАБОТНИКОВ"\*/

CREATE OR REPLACE PROCEDURE WORKERS\_DELETE(P\_ID IN INT)

IS

BEGIN

DELETE FROM WORKERS

WHERE

ID = P\_ID;

COMMIT;

END;

/\*СОЗДАНИЕ ПРОЦЕДУРЫ ИЗМЕНЕНИЕ ДЛЯ ТАБЛИЦЫ "РАБОТНИКОВ"\*/

CREATE OR REPLACE PROCEDURE WORKERS\_UPDATE(SURNAME IN VARCHAR, NAME IN VARCHAR, PATRONYMIC IN VARCHAR, PASSPORT\_NUMBER IN INT, THE\_PASSPORT\_SERIES IN INT,RESIDENCE IN VARCHAR, TELEPHONE IN VARCHAR , ID\_POSITIONS IN INT)

IS

BEGIN

UPDATE WORKERS SET

SURNAME = SURNAME,

NAME = NAME,

PATRONYMIC = PATRONYMIC,

PASSPORT\_NUMBER = PASSPORT\_NUMBER,

THE\_PASSPORT\_SERIES = THE\_PASSPORT\_SERIES,

RESIDENCE = RESIDENCE,

TELEPHONE = TELEPHONE,

ID\_POSITIONS = ID\_POSITIONS

WHERE

ID = P\_ID;

COMMIT;

END;

/\*СОЗДАНИЕ ПРОЦЕДУРЫ ДОБАВЛЕНИЯ ДЛЯ ТАБЛИЦЫ "ДОЛЖНОСТИ"\*/

CREATE OR REPLACE PROCEDURE POSITIONS\_INSERT( SALARY IN INT, NAME IN VARCHAR, DUTIES IN VARCHAR, DESCRIPTION IN VARCHAR)

IS

BEGIN

INSERT INTO POSITIONS (SALARY , NAME , DUTIES , DESCRIPTION )

VALUES (SALARY , NAME , DUTIES , DESCRIPTION );

COMMIT;

END;

/\*СОЗДАНИЕ ПРОЦЕДУРЫ УДАЛЕНИЕ ДЛЯ ТАБЛИЦЫ "ДОЛЖНОСТИ"\*/

CREATE OR REPLACE PROCEDURE POSITIONS\_DELETE(P\_ID IN INT)

IS

BEGIN

DELETE FROM POSITIONS

WHERE

ID = P\_ID;

COMMIT;

END;

/\*СОЗДАНИЕ ПРОЦЕДУРЫ ИЗМЕНЕНИЕ ДЛЯ ТАБЛИЦЫ "ДОЛЖНОСТИ"\*/

CREATE OR REPLACE PROCEDURE POSITIONS\_UPDATE(SALARY IN INT, NAME IN VARCHAR, DUTIES IN VARCHAR, DESCRIPTION IN VARCHAR)

IS

BEGIN

UPDATE POSITIONS SET

SALARY = SALARY,

NAME = NAME,

DUTIES = DUTIES,

DESCRIPTION = DESCRIPTION

WHERE

ID = P\_ID;

COMMIT;

END;

/\*СОЗДАНИЕ ПРОЦЕДУРЫ ДОБАВЛЕНИЯ ДЛЯ ТАБЛИЦЫ "ПРАЙС ЛИСТ"\*/

CREATE OR REPLACE PROCEDURE PRICE\_LIST\_INSERT( NAME\_OF\_OPERATION IN VARCHAR, PRICE IN INT, ID\_PRODUCT\_IN\_STOCK IN INT)

IS

BEGIN

INSERT INTO PRICE\_LIST (NAME\_OF\_OPERATION , PRICE , ID\_PRODUCT\_IN\_STOCK )

VALUES (NAME\_OF\_OPERATION , PRICE , ID\_PRODUCT\_IN\_STOCK );

COMMIT;

END;

/\*СОЗДАНИЕ ПРОЦЕДУРЫ УДАЛЕНИЕ ДЛЯ ТАБЛИЦЫ "ПРАЙС ЛИСТ"\*/

CREATE OR REPLACE PROCEDURE PRICE\_LIST\_DELETE(P\_ID IN INT)

IS

BEGIN

DELETE FROM PRICE\_LIST

WHERE

ID = P\_ID;

COMMIT;

END;

/\*СОЗДАНИЕ ПРОЦЕДУРЫ ИЗМЕНЕНИЕ ДЛЯ ТАБЛИЦЫ "ПРАЙС ЛИСТ"\*/

CREATE OR REPLACE PROCEDURE PRICE\_LIST\_UPDATE(NAME\_OF\_OPERATION IN VARCHAR, PRICE IN INT, ID\_PRODUCT\_IN\_STOCK IN INT)

IS

BEGIN

UPDATE PRICE\_LIST SET

NAME\_OF\_OPERATION = NAME\_OF\_OPERATION,

PRICE = PRICE,

ID\_PRODUCT\_IN\_STOCK = ID\_PRODUCT\_IN\_STOCK

WHERE

ID = P\_ID;

COMMIT;

END;

/\*СОЗДАНИЕ ПРОЦЕДУРЫ ДОБАВЛЕНИЯ ДЛЯ ТАБЛИЦЫ "ТОВАРЫ НА СКЛАДЕ"\*/

CREATE OR REPLACE PROCEDURE PRODUCT\_IN\_STOCK\_INSERT( NAME IN VARCHAR, MANUFACTURER IN VARCHAR , DESCRIPTION IN VARCHAR, PURCHASE\_VALUE IN INT,COST\_FOR\_SALE IN INT,COUNT IN INT,DO\_I\_NEED\_TO\_RE\_ORDER IN VARCHAR)

IS

BEGIN

INSERT INTO PRODUCT\_IN\_STOCK (NAME , MANUFACTURER , DESCRIPTION , PURCHASE\_VALUE ,COST\_FOR\_SALE ,COUNT ,DO\_I\_NEED\_TO\_RE\_ORDER )

VALUES (NAME , MANUFACTURER , DESCRIPTION , PURCHASE\_VALUE ,COST\_FOR\_SALE ,COUNT ,DO\_I\_NEED\_TO\_RE\_ORDER );

COMMIT;

END;

/\*СОЗДАНИЕ ПРОЦЕДУРЫ УДАЛЕНИЕ ДЛЯ ТАБЛИЦЫ "ТОВАРЫ НА СКЛАДЕ"\*/

CREATE OR REPLACE PROCEDURE PRODUCT\_IN\_STOCK\_DELETE(P\_ID IN INT)

IS

BEGIN

DELETE FROM PRODUCT\_IN\_STOCK

WHERE

ID = P\_ID;

COMMIT;

END;

/\*СОЗДАНИЕ ПРОЦЕДУРЫ ИЗМЕНЕНИЕ ДЛЯ ТАБЛИЦЫ "ТОВАРЫ НА СКЛАДЕ"\*/

CREATE OR REPLACE PROCEDURE PRODUCT\_IN\_STOCK\_UPDATE(NAME IN VARCHAR, MANUFACTURER IN VARCHAR , DESCRIPTION IN VARCHAR, PURCHASE\_VALUE IN INT,COST\_FOR\_SALE IN INT,COUNT IN INT,DO\_I\_NEED\_TO\_RE\_ORDER IN VARCHAR)

IS

BEGIN

UPDATE PRODUCT\_IN\_STOCK SET

NAME = NAME,

MANUFACTURER = MANUFACTURER,

DESCRIPTION = DESCRIPTION,

PURCHASE\_VALUE = PURCHASE\_VALUE,

COST\_FOR\_SALE = COST\_FOR\_SALE,

DO\_I\_NEED\_TO\_RE\_ORDER = DO\_I\_NEED\_TO\_RE\_ORDER

WHERE

ID = P\_ID;

COMMIT;

END;

## Функции базы данных

В данном разделе предоставлен скрипт базы данных для функций.

/\*СОЗДАНИЕ ФУНКЦИИ ПОДСЧЕТА ОБЩЕГО КОЛИЧЕСТВА КЛИЕНТОВ\*/

CREATE OR REPLACE FUNCTION COUNT\_OF\_CLIENTS RETURN INT IS ANSWERS INT;

BEGIN

SELECT COUNT (ID) INTO ANSWERS FROM CLIENTS;

RETURN ANSWERS;

END;

SELECT COUNT\_OF\_CLIENTS FROM CLIENTS;

/\*СОЗДАНИЕ ФУНКЦИИ ПОДСЧЕТА ОБЩЕГО КОЛИЧЕСТВА СОТРУДНИКОВ\*/

CREATE OR REPLACE FUNCTION COUNT\_OF\_WORKERS RETURN INT IS ANSWERS INT;

BEGIN

SELECT COUNT (ID) INTO ANSWERS FROM WORKERS;

RETURN ANSWERS;

END;

SELECT COUNT\_OF\_WORKERS FROM WORKERS;

/\*СОЗДАНИЕ ФУНКЦИИ ПОДСЧЕТА ОБЩЕГО КОЛИЧЕСТВА ЗАКАЗОВ\*/

CREATE OR REPLACE FUNCTION COUNT\_OF\_ORDERS RETURN INT IS ANSWERS INT;

BEGIN

SELECT COUNT (ID) INTO ANSWERS FROM ORDERS;

RETURN ANSWERS;

END;

SELECT COUNT\_OF\_CLIENTS FROM ORDERS;

## Представления базы данных

В данном разделе предоставлен скрипт базы данных для представлений.

/\*СОЗДАНИЕ ПРЕДСТАВЛЕНИЯ С ТАБЛИЦАМИ: ЗАКАЗЫ, КЛИЕНТЫ,ПРАЙС ЛИСТ,РАБОТНИКИ\*/

CREATE OR REPLACE VIEW O\_W\_C\_P

AS

SELECT ORD.DISCRIPTION , ORD.SPARE\_PARTS\_FOR\_REPAIRS , ORD.CAR\_NUMBER, ORD.CAR\_MODEL, ORD.FINNALY\_COST , ORD.SERVICE\_START\_DATE , C.TELEPHONE, C.NAME , P.NAME\_OF\_THE\_OPERATION,P.PRICE

FROM ORDERS ORD, WORKERS W, CLIENTS C, PRICE\_LIST P WHERE W.ID = ORD.ID\_WORKER AND C.ID = ORD.ID\_OWNER ;

SELECT\*FROM O\_W\_C\_P;

/\*СОЗДАНИЕ ПРЕДСТАВЛЕНИЯ С ТАБЛИЦАМИ: ПРАЙС ЛИСТ,ТОВАРЫ НА СКЛАДЕ\*/

CREATE OR REPLACE VIEW PRICE\_LIST\_PRODUCTS\_IN\_STOCK

AS

SELECT P.NAME\_OF\_THE\_OPERATION,P.PRICE , PR.NAME,PR.DESCRIPTION,PR.MANUFACTURER

FROM PRODUCTS\_IN\_STOCK PR, PRICE\_LIST P WHERE PR.ID = P.ID\_PRODUCT\_IN\_STOCK;

SELECT\*FROM PRICE\_LIST\_PRODUCTS\_IN\_STOCK;

/\*СОЗДАНИЕ ПРЕДСТАВЛЕНИЯ С ТАБЛИЦАМИ: РАБОТНИКИ, ДОЛЖНОСТИ\*/

CREATE OR REPLACE VIEW WORKERS\_POSITIONS

AS

SELECT W.NAME, W.RESIDENCE,W.TELEPHONE , P.SALARY, P.DUTIES,P.DESCRIPTION

FROM WORKERS W, POSITIONS P WHERE P.ID = W.ID\_POSITIONS;

SELECT\*FROM WORKERS\_POSITIONS;

# 

# ТЕСТИРОВАНИЕ

## Сценарии

### Таблицы

В данном пункте необходимо указаны таблицы с реляционными моделями для каждой из таблиц и тесты.

Таблица 3 – Реляционная модель таблицы «Clients»

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Название таблицы | Clients | | | | | | | |
| № | Название поля | | | | | | | |
| id | Surname | Name | Patronomyc | The\_passport\_series | Passport\_number | Telephone | Date of birthday |
| 1 | 6 | Алпатов | Никита | Сергеевич | 12341123 | 22313 | 89684874717 | 02.02.01 |
| 2 | 7 | Алпатов | Никита | Сергеевич | 12341123 | 22313 | 89684874717 | 02.02.01 |
| 3 | 8 | Лопин | Андрей | Антонович | 123123 | 31312312 | 89684223232 | 02.04.05 |

Таблица 4 – Реляционная модель таблицы «Positions»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название таблицы | Positions | | | | |
| № | Название поля | | | | |
| id | Salary | Name | Duties | Description |
| 1 | 1 | 20000 | Директор | Работа с документами | Рабочий день 8 часов выходные пятница и суббота |
| 2 | 2 | 20000 | Администратор | Работа с документами | Рабочий день 8 часов выходные пятница и суббота |
| 3 | 3 | 20000 | Механик | Работа над заказами | Рабочий день 8 часов выходные пятница и суббота |

Таблица 5 – Реляционная модель таблицы «Production\_in\_stock»

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Название таблицы | Products\_in\_stock | | | | | | | |
| № | Название поля | | | | | | | |
| id | Name | Manufacturer | Description | Purchase\_value | Cost\_for\_sale | Count | Do\_i\_need\_to\_re\_order |
| 1 | 1 | Двигатель | BMW | Двигатель внутреннего сгорания | 20000 | 1 | 25000 | Нет |
| 2 | 2 | Двигатель | Lada | Двигатель внутреннего сгорания | 21000 | 4 | 25000 | Нет |
| 3 | 3 | Двигатель | Reno | Двигатель внутреннего сгорания | 22000 | 6 | 25000 | Нет |

Таблица 6 – Реляционная модель таблицы «Workers»

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Название таблицы | Workers | | | | | | | | |
| № | Название поля | | | | | | | | |
| id | Surname | Name | Patronomyc | Passport\_number | The\_passport\_series | Residence | Telephone | Id\_Positions |
| 1 | 1 | Корнеев | Александр | Дмитриевич | 132423 | 23452 | Люберцы | 89684673434 | 1 |
| 2 | 2 | Патау | Александр | Генадьевич | 1324343 | 2324452 | Сочи | 89684173434 | 2 |
| 3 | 3 | Сухарев | Андрей | Генадьевич | 137423 | 238552 | Люблино | 89684673634 | 3 |

Таблица 7 – Реляционная модель таблицы «Price\_list»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название таблицы | Price\_list | | | |
| № | Название поля | | | |
| id | Name\_of\_the\_operation | Price | Id\_Product\_in\_stock |
| 1 | 1 | Замена двигателя | 30000 | 1 |
| 2 | 2 | Замена двигателя | 56000 | 2 |
| 3 | 3 | Замена двигателя | 67000 | 3 |

Таблица 8 – Реляционная модель таблицы «Orders»

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Название таблицы | Orders | | | | | | | | | | |
| № | Название поля | | | | | | | | | | |
| id | Discription | Spare\_parts\_for\_repairs | Car\_number | Car\_model | Finnale\_cost | Service\_start\_date | Do\_i\_need\_to\_re\_order | Id\_Worker | Id\_Ongoing\_operation | Id\_Owner |
| 1 | 1 | Двигатель сломан | Двигатель | v345rt | Lada | 120000 | 01.01.01 | 1 | 2 | 3 | 1 |
| 2 | 2 | Двигатель сломан | Двигатель | v345rt | BMW | 150000 | 01.01.04 | 2 | 2 | 3 | 2 |
| 3 | 3 | Двигатель сломан | Двигатель | v345rt | WAZ | 12000 | 01.01.01 | 3 | 2 | 3 | 3 |

В таблице 9 представлены тесты для таблиц. Ниже, в разделе «Результаты тестирования» представлены скриншоты выполнения тестов.

Таблица 9 − Тесты таблиц.

| № | Описание теста | Пример | Команда | Результат |
| --- | --- | --- | --- | --- |
| Т-1 | Вставка элемента в таблицу с уже существующим кодом. | Таблица Positions с 3 строками первоначально. Используемые коды от 1 до 3. | INSERT INTO POSITIONS(ID, SALARY,NAME,DUTIES,DESCRIPTION) VALUES (2,20000,'УБОРЩИК','УБИРАТЬ И МЫТЬ ПОЛЫ','6-ТИ ЧАСОВОЙ РАБОЧИЙ ДЕНЬ'); | Данные вставляются с новым значением кода. Код новой записи − 4. |
| Т-2 | Вставка элемента в таблицу с кодом, который не использовался. | Таблица Positions с 6 строками первоначально. Используемые коды от 1 до 6. | INSERT INTO POSITIONS(ID, SALARY,NAME,DUTIES,DESCRIPTION) VALUES (8,20000,'УБОРЩИК','УБИРАТЬ И МЫТЬ ПОЛЫ','6-ТИ ЧАСОВОЙ РАБОЧИЙ ДЕНЬ'); | Данные вставляются с новым значением кода. Код новой записи − 7. |
| Т-3 | Вставка элемента в таблицу с внешним ключом. | Таблица Price\_list с 3 строками первоначально. Используемые коды от 1 до 3. | INSERT INTO PRICE\_LIST(NAME\_OF\_THE\_OPERATION,PRICE,ID\_PRODUCT\_IN\_STOCK) VALUES ('УСТАНОВКА ДВИГАТЕЛЯ',400000,2); | Вставка будет произведена. |
| Т-4 | Вставка числового значения в поле типа varchar. | Таблица Price\_list с 4 строками первоначально. | INSERT INTO PRICE\_LIST(NAME\_OF\_THE\_OPERATION,PRICE,ID\_PRODUCT\_IN\_STOCK) VALUES (100,400000,2); | Вставка будет произведена. |
| Т-5 | Вставка символьного значения в поле типа integer. | Таблица Price\_list с 5 строками первоначально. Используемые коды от 1 до 5. | INSERT INTO PRICE\_LIST(NAME\_OF\_THE\_OPERATION,PRICE,ID\_PRODUCT\_IN\_STOCK) VALUES ('КОКЕОКЕЛКЕ','EGWEWG',2); | Вставка не будет произведена. |
| Т-6 | Вставка символьного-числового значения в поле типа integer. | Таблица Price\_list с 5 строками первоначально. Используемые коды от 1 до 5. | INSERT INTO PRICE\_LIST(NAME\_OF\_THE\_OPERATION,PRICE,ID\_PRODUCT\_IN\_STOCK) VALUES ('КОКЕОКЕЛКЕ','EGWК2423EWG',2); | Вставка не будет произведена. |

### Хранимые процедуры

В таблице 17 представлены тесты для хранимых процедур. Ниже, в разделе «Результаты тестирования» представлены скриншоты выполнения тестов.

Таблица 10 − Тесты хранимых процедур.

| № | Описание теста | Пример | Команда | Результат |
| --- | --- | --- | --- | --- |
| Хп-1 | Тестирование процедуры вставки данных. | Таблица CLIENTS с 3 строками первоначально. | CLIENTS\_INSERT ('ЛОПИН','АНДРЕЙ','АНТОНОВИЧ',123123,31312312,'89684223232','02.04.2005'); | Данные вставляются с новым значением кода. Код новой записи − 10. |
| Хп-2 | Тестирование процедуры вставки данных. | Таблица Workers с 5 строками первоначально. | WORKERS\_INSERT ('СУХАРM', 'АНДРЕЙ', 'ВИКТОРОВИЧ',238552,137423,'ЛЮБЛИНО','89684673634',3); | Данные вставляются с новым значением кода. Код новой записи − 6. |
| Хп-3 | Тестирование процедуры обновления данных. | Таблица Positions 7 строками первоначально. | POSITIONS\_UPDATE (20000, 'АДМИНИСТРАТОР', 'РАБОТА ','РАБОЧИЙ ДЕНЬ 8 ЧАСОВ ВЫХОДНЫЕ ПЯТНИЦА И СУББОТА',2); | Обновление будет произведено. |
| Хп-4 | Тестирование процедуры обновления данных. | Таблица Price\_list с 7 строками первоначально. | PRICE\_LIST\_UPDATE ('ЗАМЕНА ДВИГАТЕЛЯ', 35000, 1,1); | Обновление будет произведено. |
| Хп-5 | Тестирование процедуры удаления данных. | Таблица Clients С 4строками Первоначально. | CLIENTS\_DELETE (10); | Удаление произведено успешно. |
| Хп-6 | Тестирование процедуры удаления данных. | Таблица WORKERS с 6 строками первоначально. | WORKERS\_DELETE (6); | Удаление произведено успешно. |

### Функции

В таблице 18 представлены тесты для функций. Ниже, в разделе «Результаты тестирования» представлены скриншоты выполнения тестов.

Таблица 11 − Тесты для функций.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Описание теста | Пример | Команда | Результат |
| Ф-1 | Тестирование функции определения количества клиентов. | Таблица CLIENTS с 3 строками первоначально. | SELECT COUNT\_OF\_CLIENTS FROM CLIENTS; | 3 |
| Ф-2 | Тестирование процедуры определения количества сотрудников. | Таблица WORKERS с 5 строками первоначально. | SELECT COUNT\_OF\_WORKERS FROM WORKERS; | 5 |
| Ф-3 | Тестирование процедуры определения количества заказов. | Таблица ORDERS с 3 строками первоначально. | SELECT COUNT\_OF\_ORDERS FROM ORDERS; | 3 |

### Представления

В таблице 19 представлены тесты для представлений. Ниже, в разделе «Результаты тестирования» представлены скриншоты выполнения тестов.

Таблица 12 − Тесты для представлений.

| № | Описание теста | Пример | Команда | Результат |
| --- | --- | --- | --- | --- |
| П-1 | Тест вывода полной информации о заказе и сотруднике, который над ним работает. | Таблицы и поля:  ORDERS(DISCRIPTION , SPARE\_PARTS\_FOR\_REPAIRS , CAR\_NUMBER, CAR\_MODEL, FINNALY\_COST , SERVICE\_START\_DATE)  WORKERS(NAME , TELEPHONE)  CLIENTS(NAME)  POSITIONS(NAME\_OF\_THE\_OPERATION, PRICE) | SELECT\*FROM O\_W\_C\_P; | 3 |
| П-2 | Тест вывода информации о прайс-листе и товарах использующихся при определенной услуге. | Участвующие таблицы (поля таблиц):  PRICE\_LIST(NAME\_OF\_THE\_OPERATION, PRICE)  PRODUCTS\_IN\_STOCK(NAME,DESCRIPTION,MANUFACTURER) | SELECT\*FROM PRICE\_LIST\_PRODUCTS\_IN\_STOCK; | 5 |
| П-3 | Тест вывода информации о работниках и занимаемыми ими должностях. | Участвующие таблицы (поля таблиц):  POSITIONS(NAME , SALARY, DUTIES, DESCRIPTION)  WORKERS(NAME, RESIDENCE, TELEPHONE) | SELECT\*FROM WORKERS\_POSITIONS; | 5 |

## Результаты тестирования

В данном подразделе на основании каждого из разработанного сценария предоставлен результат теста.

### Таблицы

Результат теста Т-1.

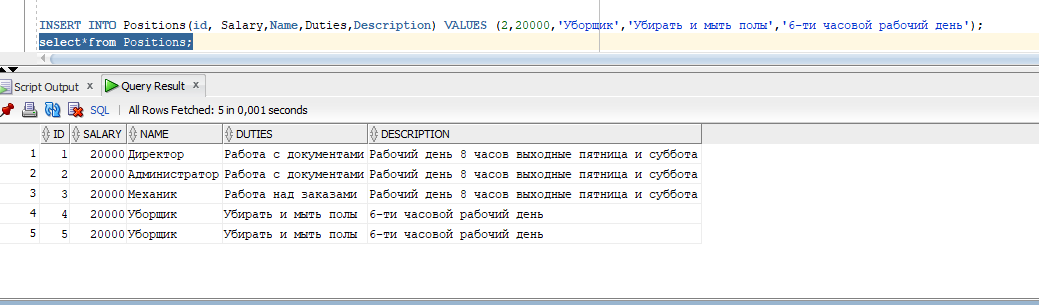


Рисунок − Результат теста Т-1.

Результат теста Т-2.

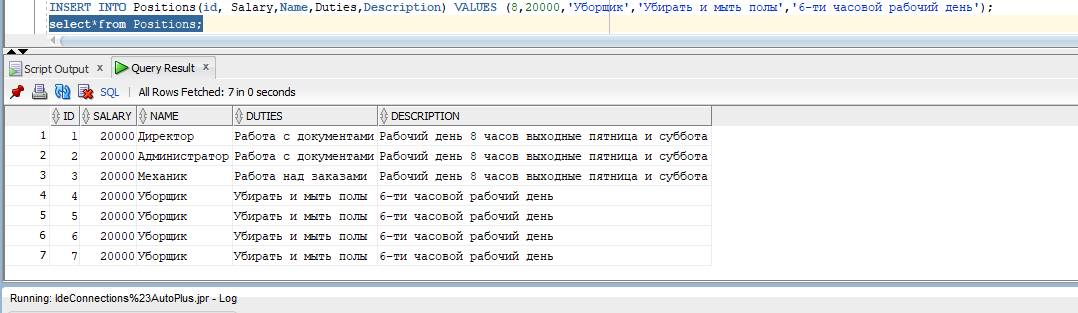


Рисунок − Результат теста Т-2.

Результат теста Т-3.

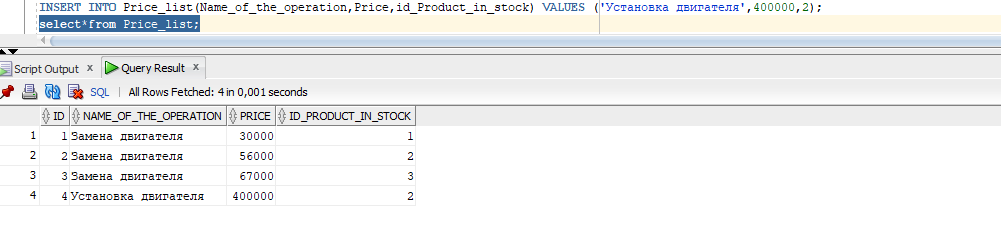


Рисунок − Результат теста Т-3.

Результат теста Т-4.

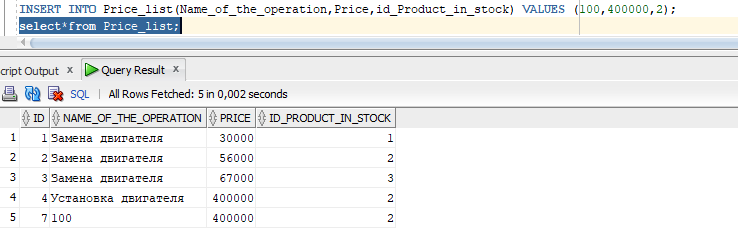


Рисунок − Результат теста Т-4.

Результат теста Т-5.

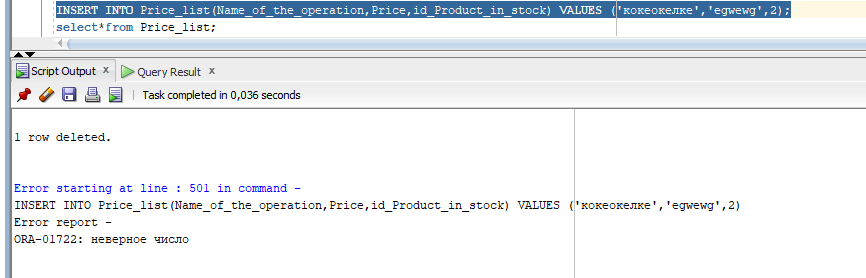


Рисунок – Результат Т-5.

Результат теста Т-6.

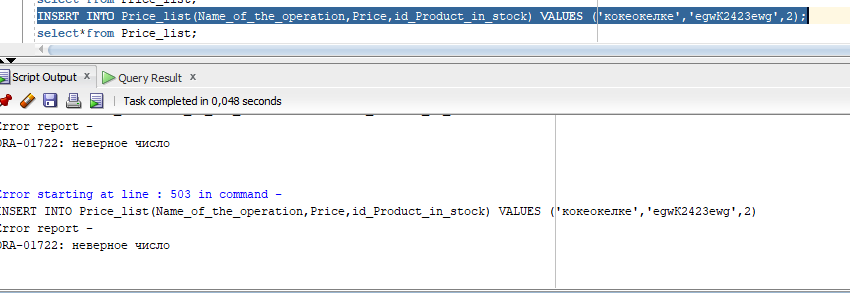


Рисунок – Результат Т-6.

### Хранимые процедуры

Результат теста Хп-1.

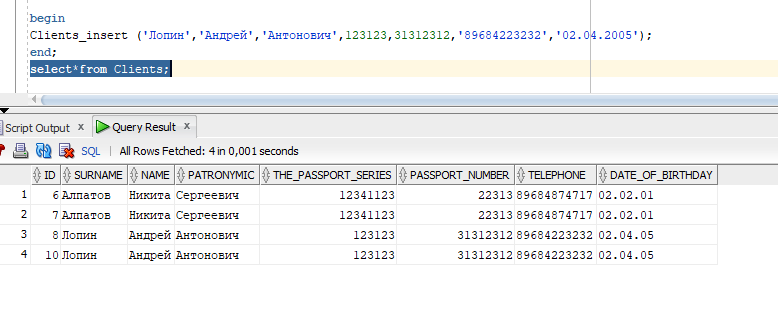


Рисунок − Результат теста Хп-1.

Результат теста Хп-2.

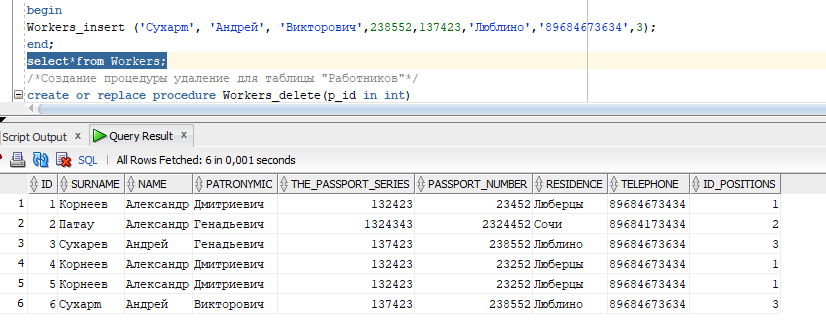


Рисунок − Результат теста Хп-2.

Результат теста Хп-3.

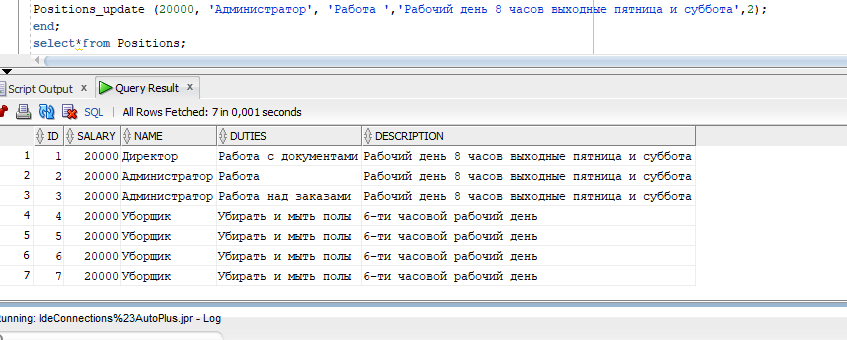


Рисунок − Результат теста Хп-3.

Результат теста Хп-4.

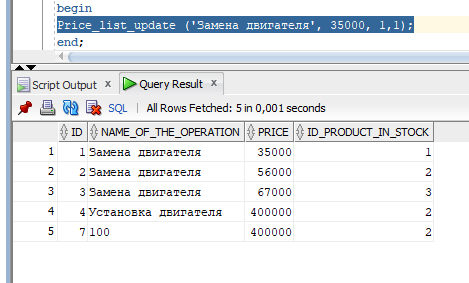


Рисунок − Результат теста Хп-4.

Результат теста Хп-5.

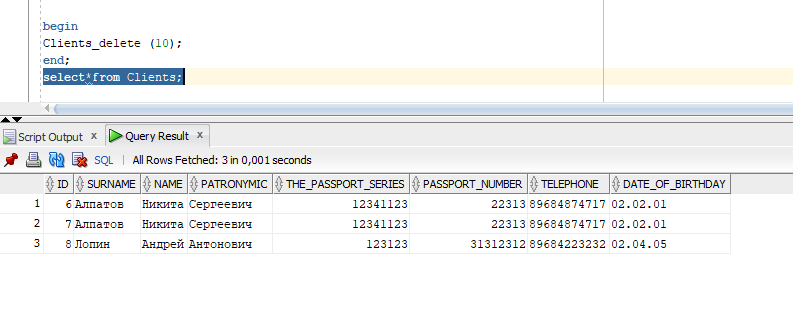


Рисунок − Результат теста Хп-5.

Результат теста Хп-6.

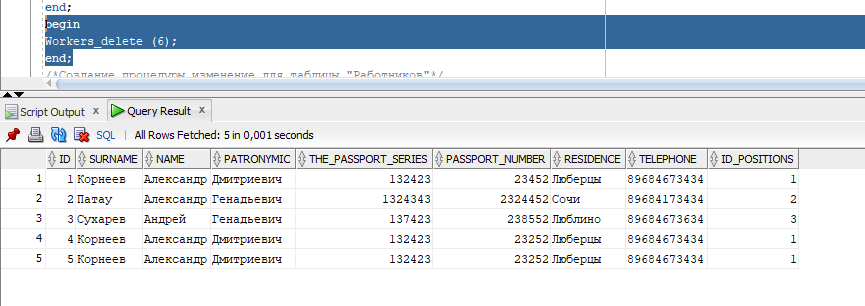


Рисунок − Результат теста Хп-6.

### Функции

Результат теста Ф-1.

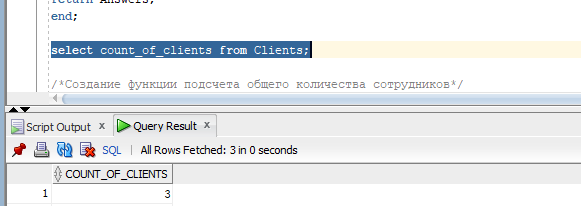


Рисунок − Результат теста Ф-1.

Результат теста Ф-2.

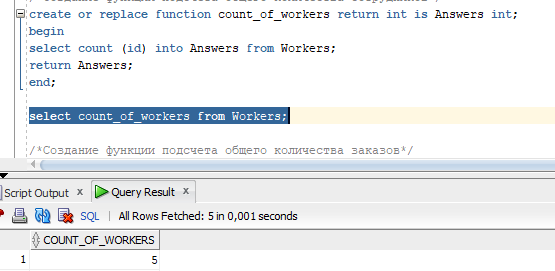


Рисунок − Результат теста Ф-2.

Результат теста Ф-3.

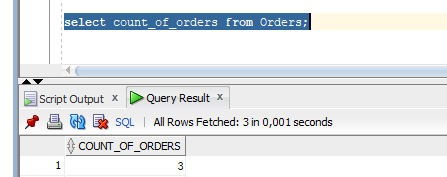


Рисунок − Результат теста Ф-3.

### Представления

Результат теста П-1.

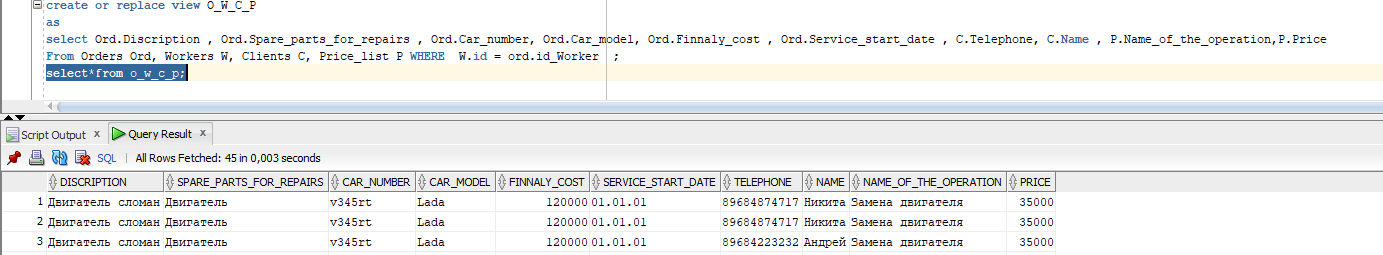


Рисунок − Результат теста П-1.

Результат теста П-2.

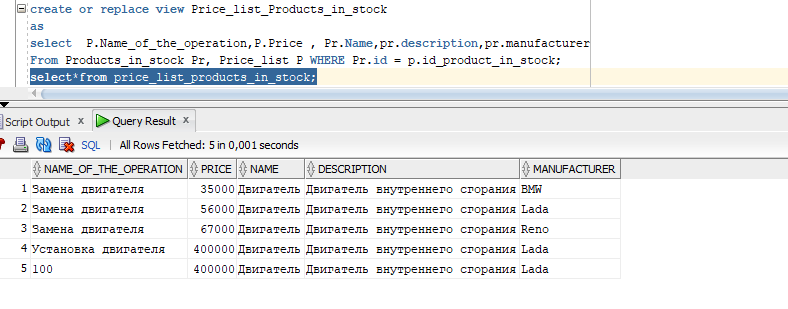


Рисунок − Результат теста П-2.

Результат теста П-3.

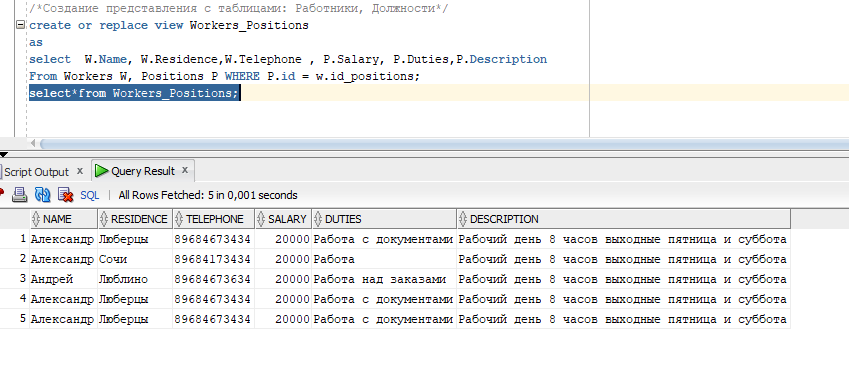


Рисунок − Результат теста П-3.

# ЗАКЛЮЧЕНИЕ

В рамках курса МДК 02.02 Технология, разработки и защиты базы данных были изучены основы работ с базой данных в СУБД Oracle SqlDeveloper, были получены знания касательно работы с представлениями, функциями, процедурами, триггерами. А также получены знания о синтаксисе языка применяющегося в Oracle SqlDeveloper. Помимо этого была реализована база данных полностью удовлетворяющая постеленные требования следовательно были получены все необходимые знания и навыки для выполнения данной работы.

# СПИСОК ИСПОЛЬЗУЕМЫХ МАТЕРИАЛОВ

В данном разделе указаны все источники, которые были использованы при разработке структуры базы данных.

1) Работа с базами данных Oracle PL/SQL, MySQL, MariaDB, SQL Server, SQLite. <https://oracleplsql.ru/> Дата обращения: 25.10.2020 г.

2) Интернет-магазин «ДискиПлюс». <https://www.diskiplus.ru/catalog/wheels/> Дата обращения: 25.10.2020 г.

3) Интернет-магазин Ozon. <https://www.ozon.ru/category/motornye-masla-8517/> Дата обращения: 25.10.2020 г.

4) Проверка авто по VIN-коду. <https://carsvin.ru/auto/nomer/> Дата обращения: 25.10.2020 г.

5) Продажа автомобилей. <https://auto.ru/moskva/cars/> Дата обращения: 25.10.2020 г.